Printed by EAST

UserID: gdistefano
Computer: WS10764
Date: 04/21/2008
Time: 12:35

Document Listing

Document	Image pages	Text pages	Error pages
JP 09171460 A	7	0	0
Total	**7**	**0**	**0**

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－171460

(43)公開日　平成９年(1997)６月30日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｇ０６Ｆ	9/06	５５０		Ｇ０６Ｆ	9/06	５５０Ｚ	
		４１０				４１０Ｂ	
	12/14	３１０			12/14	３１０Ｚ	

審査請求　未請求　請求項の数７　ＯＬ　（全 ７ 頁）

(21)出願番号　特願平7－331481

(22)出願日　平成７年(1995)12月20日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　吉田　健一
埼玉県比企郡鳩山町赤沼2520番地　株式会社日立製作所基礎研究所内

(74)代理人　弁理士　小川　勝男

(54)【発明の名称】　計算機の診断システム

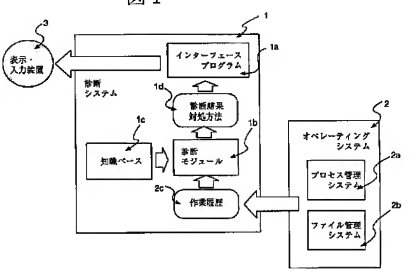
(57)【要約】

【課題】　従来、ウィルス・プログラムの感染には、ウィルス・プログラムの持つプログラムパターンと、計算機内部のファイルの内容を比較し、感染の有無を判定するウィルス検査プログラムがあった。このような従来技術では、プログラムパターンを検査プログラムが判定できないように暗号化する技術を利用したウィルス・プログラムの検査は困難であった。また、インストールミスの判断は熟練した専門家の援助が必要であった。

【解決手段】　プログラムの正常時の動作仕様や、ウィルス・プログラム等に感染した場合のプログラムの典型的動作、インストール・ミスがある場合のプログラムの動作を記憶した知識ベース１Ｃと、計算機内部の状態を観測する作業履歴２Ｃを出力する仕組みを用意しておき、記憶された動作と計算機内部の実際の状態とを比較する。

【効果】　比較結果に従い、ウィルス・プログラム等の感染やインストール・ミスを検査できる。

図１



1

【特許請求の範囲】
【請求項１】プログラムの正常時の動作仕様を記憶したデータベースと計算機内部の状態を観測する仕組みを持ち、正常時の動作仕様と計算機内部の状態とを比較することにより、ウィルス・プログラム等の障害を検知する仕組みを持つことを特徴とする計算機の診断システム。
【請求項２】プログラムの正常時の動作仕様を記憶したデータベースと計算機内部の状態を観測する仕組みを持ち、正常時の動作仕様と計算機内部の状態とを比較することにより、プログラムのインストール・ミス等を診断する仕組みを持つことを特徴とする計算機の診断システム。
【請求項３】ウィルス・プログラム等に感染した場合のプログラムの動作を記憶したデータベースと計算機内部の状態を観測する仕組みを持ち、ウィルス・プログラム等に感染した場合のプログラムの動作と計算機内部の状態とを比較することにより、ウィルス・プログラム等の障害を検知する仕組みを持つことを特徴とする計算機の診断システム。
【請求項４】インストール・ミスがある場合のプログラムの動作を記憶したデータベースと計算機内部の状態を観測する仕組みを持ち、インストール・ミスがある場合のプログラムの動作と計算機内部の状態とを比較することにより、プログラムのインストール・ミス等を診断する仕組みを持つことを特徴とする計算機の診断システム。
【請求項５】上記請求項１乃至４のいずれかに記載の計算機の診断システムを有し、計算機が正常でない動作を開始した場合に、その動作無効にする仕組みを持つことを特徴とする計算機システム。
【請求項６】計算機内部の状態を観測する仕組みを持ち、プログラムの正常時や異常時の動作を記憶した知識ベースを作成する機能を持つことを特徴とする計算機システム。
【請求項７】計算機内部の状態を観測しプログラムの正常時や異常時の動作を記憶した知識ベースを作成するために、プログラム間のファイルの入出力関係等構造的な情報まで含めて統計量等を解析し、解析結果を基に知識ベースを作成することを特徴とする請求項６項記載の計算機システム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は計算機の障害検知システムに係わり、特に従来は利用されていなかったプログラム動作に関する情報、すなわち各プログラムの関連プログラム呼出動作やファイル入出力動作を解析することにより、ウィルス・プログラムやプログラムのインストール・ミス等の障害を検知する仕組みに関する。
【０００２】
【従来の技術】従来、ウィルス・プログラムの感染は、

2

主としてウィルス・プログラムの持つプログラムパターンと、計算機内部のファイルの内容を比較し、感染の有無を判定するウィルス検査プログラムがあった。また、プログラムのインストールミスは主として人間が計算機の動作から診断を下していた。
【０００３】また、類似技術として、外部からの侵入者を発見するために、計算機の挙動を解析する技術(例えば "Detecting Intruders in Computer Systems", Teresa F.Lunt, 1993 Conference on Auditing and Computer Technologyに述べられているNIDESシステム)もあった。
【０００４】
【発明が解決しようとする課題】上記従来技術では、プログラムパターンを検査プログラムが判定できないように暗号化する技術を利用したウィルス・プログラムの検査は困難であった。また、インストールミスの判断は熟練した専門家の援助が必要であった。また、NIDESではCPUの負荷情報などを統計的に処理するため、急速に害を及ぼすウィルスへの防御方法としては不十分であった。
【０００５】本発明の目的はこの問題点を解決するために、従来は利用されていなかったプログラム動作に関する情報を解析することにより、ウィルス・プログラムやプログラムのインストール・ミス等の障害を検知する仕組みを提供することにある。
【０００６】
【課題を解決するための手段】上記目的は、プログラムの正常時の動作仕様や、ウィルス・プログラム等に感染した場合のプログラムの動作、インストール・ミスがある場合のプログラムの動作を記憶したデータベースと、計算機内部の状態を観測する仕組みを用意し、記憶された動作と計算機内部の実際の状態とを比較することにより達成される。
【０００７】
【発明の実施の形態】本発明は計算機上の適当なデータベースおよびプログラムとして実現する。
【０００８】以下、本発明の１実施例を図面を参照して説明する。
【０００９】図１は、本発明を利用した計算機システムの構成図である。１はウィルス・プログラムの感染やプログラムのインストールミスを検査するための診断システムであり、表示・入力装置３を使って計算機利用者と情報をやりとりする計算機上の適当なプログラムで良い。1aは診断システム１のインターフェース・プログラムであり、やはり計算機上の適当なプログラムで良い。２は計算機のオペレーティングシステムであり、プログラムが関連プログラムを起動した情報や、プログラムが行った入出力操作に関する情報を作業履歴2cとして出力する仕組みを持ったオペレーティングシステムで良い。
【００１０】ここで、プログラムが関連プログラムを起動した情報や、プログラムが行った入出力操作に関する

情報を作業履歴2cとして出力する仕組みを持ったオペレーティングシステム２は、オペレーティングシステム内部のサブプログラムであるプロセス管理システム2aとファイル管理システム2bに必要な機能を持たせることで容易に実現可能である。例えば、近年多くの計算機で利用されているUNIXオペレーティングシステムであれば、exec，fork，link，open，close等のシステム呼出しを実現するサブルーティンに必要な機能を持たせる事で簡単に実現可能である。

【００１１】知識ベース1cは、プログラムの正常時の動作仕様や、ウィルス・プログラム等に感染した場合のプログラムの動作、インストール・ミスがある場合のプログラムの動作を記憶する計算機上のデータベースであり、診断システム１の１部である。診断モジュール1bは知識ベース1cに記憶した動作と、オペレーティングシステムが出力する作業履歴2cを比較して、ウィルス・プログラムの感染やプログラムのインストールミスを検査し、診断結果や対処方法1dを出力する計算機上のプログラムであり、診断システム１の１部である。

【００１２】ここで、診断モジュール1bは知識ベース1cに記憶したプログラムの正常時の動作仕様と作業履歴2cを比較し、両者が一致しない場合には、ウィルスに感染しているか、プログラムのインストールミスがあると判断し、診断結果1dを出力する。また、知識ベース1cに記憶したウィルス・プログラム等に感染した場合のプログラムの動作と作業履歴2cを比較し、両者が一致する場合には、ウィルスに感染していると判断し、診断結果1dを出力する。さらに、知識ベース1cに記憶したインストール・ミスがある場合のプログラムの動作と作業履歴2cを比較し、両者が一致する場合には、プログラムのインストールミスがあると判断し、診断結果1dを出力する。

【００１３】図２は、本発明を説明するための計算機内部の処理の例であり、オペレーティングシステム２は作業履歴2cとして、図２の情報を出力する。図２において計算機の利用者は、インターフェース・プログラム1aを通してアプリケーション・プログラム4a,4b,4c,4d,4eを利用し、インターフェース・プログラム1a'を通してアプリケーション・プログラム5a,5b,5c,5dを利用し、インターフェース・プログラム1a"を通してアプリケーション・プログラム6a,6b,6c,6d,6e,6fを利用している。

【００１４】図３は、本発明を説明するための、プログラムの正常動作の例であり、知識ベース1cに記憶されている情報の例である。図３は、アプリケーション・プログラム6a(emacs)によりc programのソースコードが作成されると、作成されたプログラムはアプリケーション・プログラム6b(make)が起動したアプリケーション・プログラム6e(cc)によりコンパイルされ、アプリケーション・プログラム6f(ld)により実行形式アプリケーション・プログラム5c(prog.exe)に変換されることと、作業過程で、アプリケーション・プログラム6e(cc)は入力ファイル7a(/usr/include/stdio.h等)を、アプリケーション・プログラム6f(ld)は入力ファイル7b(/usr/lib/libc.a等)を入力として使うこと、また作業用のファイルとして出力ファイル7c(/tmp/work_file)を使うことを示している。

【００１５】図４は、本発明を説明するための、プログラムがウィルスに感染した場合の動作例であり、作業履歴2cの内容を１部抽出した場合の例である。図３との違いはアプリケーション・プログラム6f(ld)の動作である。ウィルスに感染したアプリケーション・プログラム6f(ld)は/os_image 8に書き込みを試みているが、図３の正常動作例と比較すれば、この/os_image 8に書き込みが、アプリケーション・プログラム6f(ld)の本来の仕様ではなく、ウィルスの感染によるものであることは診断できる。また、このような異常な動作を起こした時に該当プログラム(この場合ケーション・プログラム6f)の動作を停止する機能をオペレーティングシステム２に持たせることにより、ウィルス等により重大な障害を発生することを予防できる。このオペレーティングシステム２の機能は、診断システム１が「異常動作」と判定した場合にオペレーティングシステム２に、その事を通知し、オペレーティングシステム２が異常原因となったプログラムを停止させることにより容易に実現できる。

【００１６】図５は、本発明を説明するための、インストールミスがあった場合のプログラムの動作例であり、作業履歴2cの内容を１部抽出した場合の例である。図３の正常動作例と比較すれば、この本来成功すべきアプリケーション・プログラム6f(ld)による/usr/lib/libc.a等入力ファイル7bの入力作業が失敗(9a)し、その結果、実行形式プログラム5c(prog.exe)の実行が失敗(9b)していることがわかる。従来このような場合、非専門家には実行形式プログラム5c(prog.exe)の実行失敗(9b)のみがわかり、本来の原因であるアプリケーション・プログラム6f(ld)による/usr/lib/libc.a等入力ファイル7bの入力の失敗(9a)まではわからなかったが、動作例を比較することにより、このようなインストールミス(この場合は必要ファイルのインストール忘れ)も診断できる。

【００１７】上記実施例においては、簡単のため、知識ベース1cにはプログラムの正常または異常な動作が記憶されているとして説明をしたが、ウィルス感染時やインストールミスの時の対処方法を同時に記憶しておく事で、それら異常に対する対処方法を出力する事もでき、本発明のその他の実施例である。

【００１８】図６は、本発明を利用して知識ベース1cに計算機動作を記憶した場合の実施例の構成図であり、1eが知識ベース作成モジュールである。本実施例では知識ベース作成モジュールが、1.ウィルスにも感染しておらずソフトも正常にインストールされている状態、および、2.知識ベース1c作成のためにウィルスを感染させた状態、および、3.知識ベース1c作成のためにインストー

ル上の不都合を生じさせた状態の計算機の典型的な動作を知識ベース1cに記憶する。

【００１９】ここで、知識ベース1cに記憶される計算機の動作は、オペレーティングシステム２から出力される作業履歴2cの適当な１部で良い。ここで、作業履歴2cはプログラム間のファイルの入出力関係を構造的な情報として図３、４、５に例示したグラフの形を持っている。このグラフ中に繰り返し現れるパターンを抽出すれば、1.ウィルスにも感染しておらずソフトも正常にインストールされている状態、および、2.知識ベース1c作成のためにウィルスを感染させた状態、および、3.知識ベース1c作成のためにインストール上の不都合を生じさせた状態、それぞれの計算機の典型的な動作を知識ベース化できる。

【００２０】グラフ中に繰り返し現れるパターンの抽出はどのような手法を用いても良いが、文献「推論過程からの概念学習（1）類型的推論過程の抽出、吉田・元田、人工知能学会誌、Vol．7，No．4，pp.119-129（1992）」に示された方法を用いれば、プログラム間のファイルの入出力関係等構造的な情報まで含めて統計量等を解析し、解析結果を基に知識ベース1cを作成することができる。

【００２１】図１と図６に例示した実施例を組み合わせ、ウィルス・プログラム等の感染やインストール・ミスの検査と、知識ベース1cの作成を同じ機械の上で行えるようにした計算機も本発明の別の実施例である。この場合、作成した知識ベース1cに、新たに発見したウィルスに固有のプログラムパターンを一緒に抽出して記憶することにより、従来手法であるプログラムのメモリ上に記憶された文字列としてのウィルスのパターンを調べる仕組みのシステムのセキュリティを向上させることもできる。すなわち、本発明を利用した計算機システムで作業中に、計算機が新種のウィルスに感染したとする。本発明によりウィルスがオペレーティングシステム２に影響を及ぼそうとしても、安全にウィルスの動きを止め、該ウィルスに関する知識ベース1cを作成できる。この時新たに発見したウィルスに固有のプログラムパターンを一緒に抽出して記憶することにより、従来手法であるプログラムのメモリ上に記憶された文字列としてのウィルスのパターンを調べる仕組みのシステム用の知識ベースも作成できる。この場合、新しいプログラムを起動する前に抽出したプログラムパターンと比較することで、該プログラムがウィルスに感染しているか否か検査できる。

【００２２】

【発明の効果】以上の実施例で明らかなように，本発明によれば、データベースに記憶された動作と、計算機内部の状態を比較し、比較結果に従い、ウィルス・プログラム等の感染やインストール・ミスを検査できる。また、計算機が正常でない動作を開始した場合に、その動作を無効にできるので、ウィルス・プログラムの感染やインストール・ミスなどによるファイルの破壊等の障害を回避することができる。

【００２３】また、プログラムのメモリ上に記憶された文字列としてのパターンでなく、動作で診断を行う為、プログラムパターンを検査プログラムが判定できないように暗号化する技術を利用したウィルス・プログラムの検査も可能である。さらに通常使用しているオペレーティングシステムに組み込まれているので、別の装置でウィルスを検査する等の準備も不要であり、不意に感染したウィルスにも対処できる。正常時の動作パターンと比較してウィルスの検査をする実施例は動作のわかっていない未知のウィルスへも対応できる。

【００２４】さらに診断用の知識ベースも自動作成でき、未知ウィルスに感染した場合、自動的に未知ウィルスに関する知識ベースを作成することもできるので、ウィルスのプログラムパターンを記憶することで、従来手法であるプログラムのメモリ上に記憶された文字列としてのウィルスのパターンを調べる仕組みのシステムのセキュリティを向上させることもできる。

【図面の簡単な説明】

【図１】本発明を利用した計算機システムの構成図。

【図２】本発明を説明するための計算機内部の処理の例。

【図３】本発明を説明するための、プログラムの正常動作の例。

【図４】本発明を説明するための、プログラムがウィルスに感染した場合の動作例。

【図５】本発明を説明するための、インストールミスがあった場合のプログラムの動作例。

【図６】本発明を利用した計算機システムの別の実施例の構成図。

【符号の説明】

1　診断システム
1a　インターフェース・プログラム
1a’　インターフェース・プログラム
1a”　インターフェース・プログラム
1b　診断モジュール
1c　知識ベース
1d　診断結果・対処方法
1e　知識ベース作成モジュール
2　オペレーティングシステム
2a　プロセス管理システム
2b　ファイル管理システム
2c　作業履歴
3　表示・入力装置
4a　アプリケーション・プログラム
4b　アプリケーション・プログラム
4c　アプリケーション・プログラム
4d　アプリケーション・プログラム

7

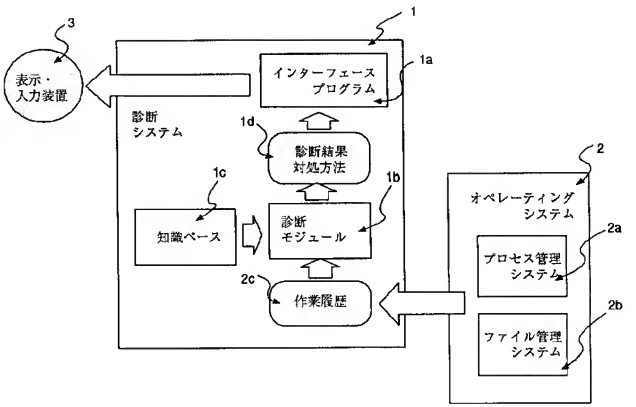
4e　アプリケーション・プログラム
5a　アプリケーション・プログラム
5b　アプリケーション・プログラム
5c　アプリケーション・プログラム
5d　アプリケーション・プログラム
6a　アプリケーション・プログラム
6b　アプリケーション・プログラム
6c　アプリケーション・プログラム
6d　アプリケーション・プログラム

8

6e　アプリケーション・プログラム
6f　アプリケーション・プログラム
7a　入力ファイル
7b　入力ファイル
7c　出力ファイル
8　出力ファイル
9a　エラー
9b　エラー。

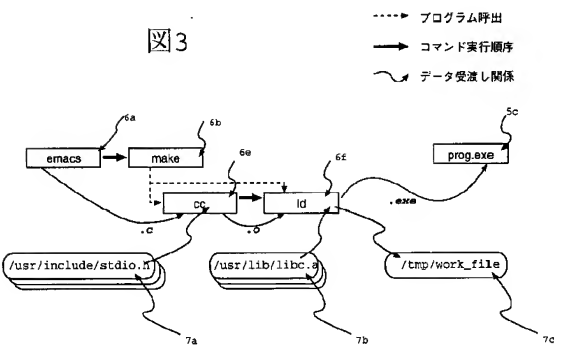
【図１】

図１



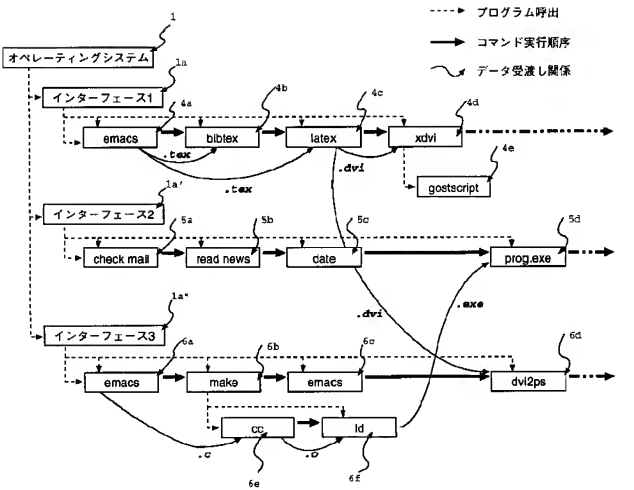
【図３】

図３



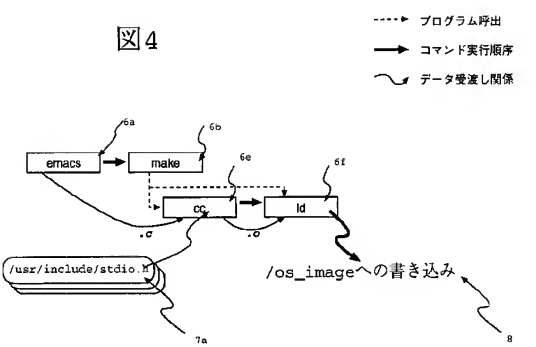
【図2】

図2


プログラム呼出
コマンド実行順序
データ受渡し関係
1
オペレーティングシステム
1a
インターフェース1
4a
emacs
4b
bibtex
4c
latex
4d
xdvi
4e
gostscript
.tex
.tex
.dvi
1a'
インターフェース2
5a
check mail
5b
read news
5c
cate
5d
prog.exe
.dvi
.exe
1a*
インターフェース3
6a
emacs
6b
make
6c
emacs
6d
dvi2ps
cc
ld
.c
.o
6e
6f

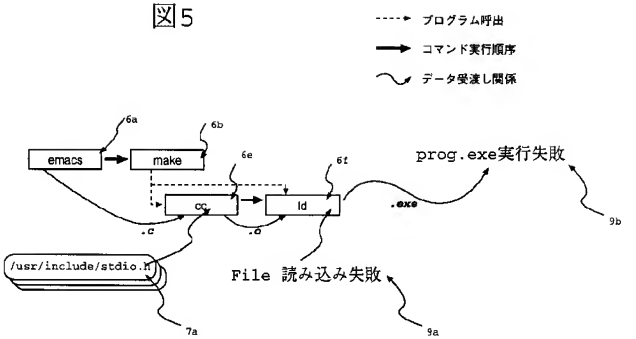
【図4】

図4


プログラム呼出
コマンド実行順序
データ受渡し関係
6a
emacs
6b
make
6e
cc
6f
ld
.c
.o
/usr/include/stdio.h
/os_imageへの書き込み
7a
8

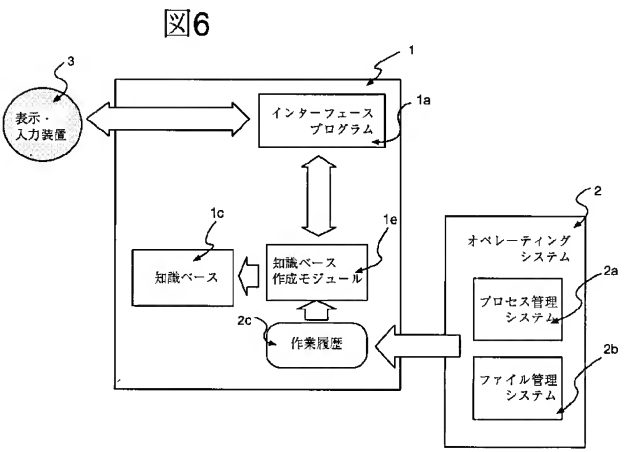
【図5】

図5


プログラム呼出
コマンド実行順序
データ受渡し関係
6a
6b
emacs
make
6e
6f
cc
ld
prog.exe実行失敗
.exe
9b
.c
.o
/usr/include/stdio.h
File 読み込み失敗
7a
9a

【図6】

図6


3
1
表示・
入力装置
インターフェース
プログラム
1a
2
1c
1e
オペレーティング
システム
知識ベース
知識ベース
作成モジュール
2a
プロセス管理
システム
2c
作業履歴
2b
ファイル管理
システム